EAST Search History

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	Time Stamp
L1	17032728	@ad<"19981008"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/07/09 11:07
L2	12	(David near2 Feldmeier).in.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/07/09 11:08
L3	0	(Tyker near2 Arnold).in.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/07/09 11:08
L4	5	(Tyler near2 Arnold).in.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/07/09 11:08
L5	11592954	2 or "5"	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/07/09 11:09
L6	11592954	2 or 5	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/07/09 11:09
L7	12	2 and 5	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/07/09 11:09
L8	630	711/108.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/07/09 11:09
L9	29798	"711"/$.ccls.	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/07/09 11:17

EAST Search History

L10	6819	content adj addressable adj memory	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/07/09 11:20
L11	426480	CAM	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/07/09 11:20
L12	336	TLAT or DLAT	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/07/09 11:21
L13	2501	translation adj lookaside adj buffer	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/07/09 11:21
L14	5189	TLB	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/07/09 11:21
L15	5566	13 or 14	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/07/09 11:22
L16	6	(ternary adj hierarch$6) near3 address$4	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/07/09 11:23
L17	1	(ternary adj hierarch$6) near3 memory	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/07/09 11:23
L18	8979	(ternary or hierarch$6) near3 memory	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/07/09 11:28
L19	45366	logic$4 near3 ("AND" or "ANDed" or "and" or "anded")	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/07/09 11:27

EAST Search History

L20	340	contiguous near2 mask$4	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/07/09 11:28
L21	43	19 and 20	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/07/09 11:28
L22	428188	10 or 11	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/07/09 11:28
L23	7	21 and 22	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/07/09 11:28
L24	2539	(ternary or hierarch$6) near3 address$3	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/07/09 11:28
L25	3	23 and 24	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/07/09 11:29
L26	3	23 and 18	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/07/09 11:29
L27	0	1 and 26	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/07/09 11:29
L28	0	1 and 25	US-PGPUB; USPAT; EPO; JPO; DERWENT; IBM_TDB	OR	OFF	2006/07/09 11:29

Sun, 9 Jul 2006, 11:45:19 AM EST

Edit an existing query or compose a new query in the Search Query Display.

Select a search number (#) to:

- Add a query to the Search Query Display
- Combine search queries using AND, OR, or NOT
- Delete a search
- Run a search

Search Query Display

[redacted]

Recent Search Queries

#1	((cam or (content addressable memory))<in>metadata)
#2	(address and mask*<IN>metadata)
#3	((ternary or hierarch*) and address*<IN>metadata)
#4	(contiguous and mask*<IN>metadata)
#5	(contiguous and mask*<IN>metadata)
#6	TLB or (translation adj lookaside adj buffer)
#7	CAM and (segmenting or segmented or segments)
#8	(CAM and partition<IN>metadata)
#9	(higest CAM address<IN>metadata)
#10	(highest cam address<in>metadata)
#11	(lowest and cam and address<in>metadata)
#12	highest and cam and address
#13	(highest and cam and address) <AND> ((lowest and cam and address<in>metadata))
#14	((highest and cam and address) <AND> ((lowest and cam and address<in>metadata))) <AND> (((cam or (content addressable memory)) <in>metadata))
#15	(((highest and cam and address) <AND> ((lowest and cam and address<in>metadata))) <AND> (((cam or (content addressable memory)) <in>metadata))) <AND> ((contiguous and mask*<IN>metadata))
#16	(((ternary or hierarch*) and address*<IN>metadata)) <AND> (((highest and cam and address) <AND> ((lowest and cam and address<in>metadata))) <AND> (((cam or (content addressable memory))<in>metadata)))
#17	((((ternary or hierarch*) and address*<IN>metadata)) <AND> (((highest and cam and address) <AND> ((lowest and cam and address<in>metadata))) <AND> (((cam or (content addressable memory))<in>metadata)))) <AND> ((contiguous and mask*<IN>metadata))

[redacted]

Terms used
CAM hierarchical memory ternary address contiguous mask

Found **4** of **178,880**

Sort results by relevance

Display results expanded form

Save results to a Binder

Search Tips

Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 1 - 4 of 4

Relevance scale

1 Full papers: Tree bitmap: hardware/software IP lookups with incremental updates

Will Eatherton, George Varghese, Zubin Dittia

April 2004 **ACM SIGCOMM Computer Communication Review**, Volume 34 Issue 2

Publisher: ACM Press

Full text available: pdf(189.39 KB) Additional Information: full citation, abstract, references

Even with the significant focus on IP address lookup in the published literature as well as focus on this market by commercial semiconductor vendors, there is still a challenge for router architects to find solutions that simultaneously meet 3 criteria: scaling in terms of lookup speeds as well as table sizes, the ability to perform high speed updates, and the ability to fit into the overall memory architecture of an Level 3 forwarding engine or packet processor with low systems cost overhead. I ...

2 Scalable high speed IP routing lookups

Marcel Waldvogel, George Varghese, Jon Turner, Bernhard Plattner

October 1997 **ACM SIGCOMM Computer Communication Review , Proceedings of the ACM SIGCOMM '97 conference on Applications, technologies, architectures, and protocols for computer communication SIGCOMM '97**, Volume 27 Issue 4

Publisher: ACM Press

Full text available: pdf(1.66 MB) Additional Information: full citation, abstract, references, citings, index terms

Internet address lookup is a challenging problem because of increasing routing table sizes, increased traffic, higher speed links, and the migration to 128 bit IPv6 addresses. IP routing lookup requires computing the best matching prefix, for which standard solutions like hashing were believed to be inapplicable. The best existing solution we know of, BSD radix tries, scales badly as IP moves to 128 bit addresses. Our paper describes a new algorithm for best matching prefix using binary search o ...

3 Scalable high-speed prefix matching

Marcel Waldvogel, George Varghese, Jon Turner, Bernhard Plattner

November 2001 **ACM Transactions on Computer Systems (TOCS)**, Volume 19 Issue 4

Publisher: ACM Press

Full text available: pdf(933.02 KB) Additional Information: full citation, abstract, references, citings, index terms

Finding the longest matching prefix from a database of keywords is an old problem with a

number of applications, ranging from dictionary searches to advanced memory management to computational geometry. But perhaps today's most frequent best matching prefix lookups occur in the Internet, when forwarding packets from router to router. Internet traffic volume and link speeds are rapidly increasing; at the same time, a growing user population is increasing the size of routing tables against which p ...

Keywords: collision resolution, forwarding lookups, high-speed networking

4 Fowarding: Longest prefix matching using bloom filters

Sarang Dharmapurikar, Praveen Krishnamurthy, David E. Taylor

August 2003 **Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications**

Publisher: ACM Press

Full text available: pdf(207.32 KB) Additional Information: full citation, abstract, references, index terms

We introduce the first algorithm that we are aware of to employ Bloom filters for Longest Prefix Matching (LPM). The algorithm performs parallel queries on Bloom filters, an efficient data structure for membership queries, in order to determine address prefix membership in sets of prefixes sorted by prefix length. We show that use of this algorithm for Internet Protocol (IP) routing lookups results in a search engine providing better performance and scalability than TCAM-based approaches. The ke ...

Keywords: IP lookup, forwarding, longest prefix matching

Results 1 - 4 of 4